

REMARKSI. Introduction

In response to the Office Action dated September 18, 2003, claims 31-36 have been added. Claims 1-36 remain in the application. Re-examination and re-consideration of the application, as amended, is requested.

II. Prior Art Rejections

On page (2) of the Office Action, claims 1, 11, and 21 were rejected under 35 U.S.C. §102(e) as being anticipated by Benson et al., U.S. Patent No. 5,845,281 (Benson). On page (3) of the Office Action, claims 2, 12, and 22 were rejected under 35 U.S.C. §103(a) as being unpatentable over Benson in view of Dougherty et al., "The Mosaic Handbook for the X Window System," (Dougherty). On page 4 of the Office Action, claims 3, 6-10, 13, 16-20, 23, and 26-30 were rejected under 35 U.S.C. §103(a) as being unpatentable over Benson. On page (9) of the Office Action, claims 4-5, 14-15, and 24-25 were rejected under 35 U.S.C. §103(a) as being unpatentable over Benson in view of Oliver, "Netscape 2 Unleashed, 1996, sams.net publishing, p. 360," (Oliver).

Specifically, claim 1 was rejected as follows:

In regard to Independent claim 1 (and similarly Claims 11 and 21), Benson et al. teaches managing a data object so as to comply with predetermined conditions for usage of the data object. To control the usage of the data object, a set of control data defining the uses of the data object is created (Abstract, lines 1-4; compare with Claim 1, 11 and 21 "A method for utilizing an object that is dependent on object data comprising:"). Benson et al. teaches that a copy of the data object is concatenated with the user set of control data (Col. 3, lines 33-34) Benson et al. Further teaches that concatenation of the file and object data does not imply that the object data and the object are together, but rather that the object data does not reside inside the file (Col. 3, lines 32-38, Figs. 11, 12a,b). Furthermore, the control data can be stored as one or more separate files (Col. 7, lines 1-2; compare with Claim 1, 11 and 21 "storing the object data for the object separate from a file containing an instance of the object"). Benson et al. Teaches that a user may request authorization for usage (one such use would be to display the file) of the data object residing at a data provider's processor via a data network or in any other appropriate way (Col 3 lines 45-47; compare with Claims 1, 11 and 21 "obtaining a request to load the file"). Benson et al. teaches that before the data package (containing the file and object data) is transferred to the user, it should be confirmed that the request for authorization for usage has been granted (Col. 3, lines 63-66; compare with Claims 1, 11 and 21 "determining if the object data is available"). Benson et al. teaches that once the data package has been obtained, it is read by user software, decrypted if necessary, and is then able to be displayed by the user software (Col. 4, lines 9-22, Col 13, lines 14-28; compare with Claims 1, 11 and 21 "obtaining the object data and utilizing the object data to display a graphical representation of the object").

Applicant traverses the above rejections for one or more of the following reasons:

- (1) Neither Benson, Dougherty, nor Oliver teach, disclose or suggest storing object data for an object separate from a file containing an instance of the object;
- (2) Neither Benson, Dougherty, nor Oliver teach, disclose or suggest determining if object data is available;
- (3) Neither Benson, Dougherty, nor Oliver teach, disclose or suggest obtaining the object data if available; and
- (4) Neither Benson, Dougherty, nor Oliver teach, disclose or suggest using object data (if available) to display a graphical representation of the object.

Independent claims 1, 11, and 21 are generally directed to the use and storage of objects. Specifically, the claims provide that data for an object is stored separately from the location (i.e., the file) where an instance of the object is stored. The file containing the instance of the object is loaded. Upon loading, the invention determines if the object data (that is stored separately) is available. If the object data is available, the object data is obtained (i.e., from the separate location). Further, the obtained object data is used to display a graphical representation of the object.

The cited references do not teach nor suggest these various elements of Applicant's independent claims. The Office Action relies on Benson to teach all of the aspects of the claimed invention. Benson describes two distinct aspects: (1) a data object, and (2) control data. An author creates the data object and conditions for using the data object (referred to as the control data) (see col. 5, lines 16-23). This control data that controls the usage of the data object may be stored in a header file and a usage data file (see col. 7, lines 1-2). Benson also suggests that the control data does not reside in the data object, but outside of it, thereby allowing usage control independently of the data object format (see col. 3, lines 32-38). When the user wants to use the data object, a special user program checks whether the usage complies with the control data. If so, the usage is enabled. Otherwise it is disabled. (See Abstract).

However, Benson lacks any discussion about object instances and the storage of an object separate from an instance of the object itself. Applicant assumes that the Office Action is equating the claimed object data to Benson's data object. However, the mere use of the term "data object" by Benson serves to distinguish Benson from the claimed object data. Benson's data object is an object itself. Applicant also notes that object instances are not described anywhere in Benson. In fact, an electronic search of Benson for the term "instance" provides no relevant results.

Nonetheless, in an attempt to apply Benson to the presently claimed invention, Applicant has tried to determine what the use of Benson's terms means. The only possible scenario is that Benson's "data object" is similar to Applicant's combined object data and object instance.

Applicant uses the term "object data" which is further described in the specification (see page 8, lines 12-21). Previous dependent claims 10, 20, and 30, and new dependent claims 31-36 provide further details on what the object data comprises. Essentially, object data is data that may represent the graphical representation of a component, content of a document (e.g., the actual text of a document), or formatting information. The object instance/component is dependent on the data such that the instance/component must have access to the data in order for the component to be resurrected, used, displayed, etc.

Examining Benson, the data objects are delivered across a network and may comprise books, films, video, news, music, software, games, etc. (see col. 1, lines 33-37). Benson attempts to control the use of such data objects to ensure copy protection from piracy and royalties (see col. 1, lines 29-41). The data object is digital data, analog data, or a combination or hybrid of analog and digital data (see col. 5, lines 16-22). Thus, Benson's data object contains the data for the object itself and is the object. In other words, Benson does not separate the object data from the instance of the object itself. To control the use of the data object, Benson provides for control data that is packaged together with the data object. However, as stated above, the control data merely provides the ability to control the usage of the data object (see col. 6, lines 24-27). The control data may be stored in a header file and a usage data file. The header file has an object identifier, that uniquely identifies the control data and/or its associated data object, a title, a format code, and a security code (see col. 7, lines 1-6). Further, the format code merely represents the format of fields in the usage data file (see col. 7, lines 6-7) (and not the format of the object itself or a component).

In order to apply Benson, various of the following equivalencies must be made: (1) Benson's control data must either be the object data OR the instance of the object, and/or (2) Benson's data object must be either the object data OR the instance of the object. Neither of these conditions/equivalencies are met. Based on the description in Benson, Benson may in fact teach that Benson's control data is not stored in the same location as Benson's data object. However, neither Benson's control data nor Benson's data object are equivalent to the claimed object data or instance of the object.

The first equivalency that must be met for Benson to teach the invention is that Benson's control data must either be equivalent to (or suggest) the object data OR the instance of the object. Benson's control data merely controls how Benson's data object is used. Accordingly, Benson's control data is not object data that an object is dependent on. In this regard, the invention's object data is described in the specification and further set forth in claims 10, 20, and 30-36. Benson does not teach the existence of such object data in Benson's control data. Further, Benson's control data is not equivalent to an instance of the object itself. Since Benson's control data only controls how Benson's data object is used, it is not an instance of the object itself.

The second equivalency that must be met for Benson to teach the invention is that Benson's data object must either be equivalent to (or suggest) the object data OR the instance of the object. The claims specifically provide that the object data is stored separately from the instance of the object. As described above, Benson does not teach for such separate storage. Instead, Benson teaches the use of a "data object" that includes all of the object data. Accordingly, Benson fails to teach such a separate storage. In this regard, Benson's data object is not equivalent to the instance of the object (that does not contain the object data). Instead, Benson's data object contains all of the data for the object. In other words, the data for Benson's data object is stored in Benson's data object itself and not separately.

Another claim element provides for utilizing the object data to display a graphical representation of the object. To teach this claim element, the Office Action first relies on col. 4, lines 9-22 which provides:

Once the data object has been packaged in the above-described manner, it can only be accessed by a user program which has built-in usage control and means for decrypting the data package. The user program will only permit usages defined as acceptable in the control data. Moreover, if the control data comprises a security control element, the security procedure prescribed therein has to be complied with. In one embodiment, the usage control may be performed as follows. If the user decides to use a data object, the user program checks the control data to see if this action is authorized. More particularly, it checks that the number of authorized usages of this kind is one or more. If so, the action is enabled and the number of authorized usages decremented by one. Otherwise, the action is interrupted by the user program and the user may or may not be given the opportunity to purchase the right to complete the action.

As can be seen by this text, there is reference, implicit or explicit to displaying a graphical representation of the object. The above text merely provides for granting access/usage to a user program in accordance with the control data. In this regard, the cited text does not even remotely refer to displaying an object whatsoever.

The Office Action also relies on col. 13, lines 14-28. Col. 13, lines 10-28 provides:

Assume that a user has found the image at an electronic bulletin board (BBS) and is interested in using it. He then loads the data package 40 containing the image to his data processor and stores it as a file in the bulk storage. The user then executes the user program 35 and requests to preview the image. The user program then performs steps 1505-1507 of the flow diagram in FIG. 15. The request for a preview of the image is compared with the data field of the usage element "code for usage type approved". In this example, the code "9" designates that previews are permitted. Thus, the requested preview is OK. Then, the user program 35 performs step 1509-1515 of FIG. 15. Since the format code "a" and the security code "b" of the header data indicate that neither conversion, nor decompression, nor security treatment is required, the user program only decrypts the object data. The usage manager module 1403 then displays the preview on the user's data processor and passes control back to the user interface 1402.

This text merely provides for loading a data package containing an image where it is stored as a file. A user program is then used to request a preview of the image. The user program evaluates the control data to determine if the user may preview the image (see col. 13, lines 10-28 and the steps of FIG. 15). If the control data provides appropriate access capabilities, the user program displays the preview. However, referring to the above stated differences, Benson does not use object data (which is stored in a different location/file from the instance of the object) to display a graphical representation of the object itself. Instead, the data object (i.e., the image) is merely displayed if the control data allows its display. Without distinguishing between the object data and instance of the object, it is impossible for Benson to teach the last claimed element of utilizing the object data to display a graphical representation of the object. Further, as claimed the object data is obtained from a different location. In Benson, even assuming the image is equivalent to the object data, it is not retrieved from a location/file that is stored separately from the instance of the object itself. Instead, the image (the entire data object image instance including the data of the data object) is merely retrieved from a bulletin board and displayed if the control data permits such a display.

The Office Action also rejects claim 2 that provides for displaying an empty graphical representation IF the object data is not available. To teach this claim element, the Office Action relies on Dougherty. Dougherty provides the ability for a user to set a Delay Image Loading option that prevents the automatic loading of inline images. Thus, if the user elects not to load an image, regardless of whether the image is available or not, Dougherty provides the ability to delay such loading. Accordingly, instead of displaying an empty representation if object data is not available, Dougherty provides the ability to delay loading an image whenever the user desires not to load such

data. Such a user option is not remotely similar to data availability as claimed. Accordingly, Applicant submits claim 2 is allowable over the cited references.

Claim 8 provides that the object data is maintained and updated by a supplier on a supplier's network. In other words, while the instance of the object is maintained in one location, the data for the object is maintained and updated by a supplier elsewhere. Claim 10 further provides that the file is a drawing, the object is a drawing component, and the object data provides the graphical representation of the drawing component. Both of these claims provide the ability for a remote supplier of an object (e.g., a toilet manufacturer such as Kohler) to maintain a list of objects (e.g., graphical objects such as toilets) that may be included in a drawing file remotely. Whenever the drawing file (that contains the instance of the object) is ready to display the toilet, the drawing program determines if the object data (i.e., the information needed to display the toilet maintained by Kohler) is available and displays it accordingly.

To reach these claims, the Office Action exclusively relies on Benson. With respect to claim 8, the Office Action relies on obviousness with respect to how a supplier would keep control of an object. However, Benson fails to describe a supplier or supplier's network. In Example 1 of Benson, the image is stored on a bulletin board for a user to retrieve and not on a supplier's network where the data object is separate from the instance of the object. Instead, everything in Example 1 is stored on the bulletin board (i.e., the secure package that comprises both the data object and the control data) (see col. 8, lines 29-col. 9, line 9). In Benson's example 2, a broker stores the data object and control data in a package together in a database of the broker. The broker can then transmit the entire package to the user. Again, there is no separate storage of both the object data and instance of the object in separate locations or networks. Instead, they are located in the same location (See col. 9, line 10-col. 13, line 5).

The differences between all of these examples and the method of Benson is that the broker or author of the material in Benson has no way to control, maintain and/or update the data object or control data once it is distributed to the user or uploaded to the bulletin board. It is packaged there and the user may download and use it in accordance with the control data in the package. Further, there is no separate storage as described above. Such limitations are in contrast to the presently claimed invention wherein the supplier may maintain and update the object data on the supplier's network.

With respect to claim 10, the Office Action states that Benson does not teach that the file is a drawing and relies on ordinary skill in the art. Such reliance is without merit. The claims provide that the file is a drawing, and the object is a drawing component (i.e., a component of the drawing). Further, the claims provide that the object data provides the graphical representation of the drawing component. In Benson, there is merely an artist's image and not a drawing having objects or drawing components. Further, Benson fails to indicate how object data that is stored separately from the object instance itself can provide a graphical representation of the drawing component (as claimed). Instead, the Office Action merely states that the user has the benefit of a graphical representation of the object. Such a statement ignores the fact that the graphical representation is stored separately from the file (i.e., the drawing) containing the instance of the object itself. Benson completely fails to teach such claim elements, implicitly or explicitly.

Moreover, the various elements of Applicant's claimed invention together provide operational advantages over Benson, Dougherty, and Oliver. In addition, Applicant's invention solves problems not recognized by Benson, Dougherty, and Oliver.

Thus, Applicant submits that independent claims 1, 11, and 21 are allowable over Benson, Dougherty, and Oliver. Further, dependent claims 2-10, 12-20, and 22-30 are submitted to be allowable over Benson, Dougherty, and Oliver in the same manner, because they are dependent on independent claims 1, 11, and 21, respectively, and thus contain all the limitations of the independent claims. In addition, dependent claims 2-10, 12-20, and 22-30 recite additional novel elements not shown by Benson, Dougherty, and Oliver.

III. Conclusion

In view of the above, it is submitted that this application is now in good order for allowance and such allowance is respectfully solicited. Should the Examiner believe minor matters still remain that can be resolved in a telephone interview, the Examiner is urged to call Applicant's undersigned attorney.

Respectfully submitted,

GATES & COOPER LLP
Attorneys for Applicant(s)

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: December 18, 2003

By: Jason S. Feldmar
Name: Jason S. Feldmar
Reg. No.: 39,187

JSF/